ISSN 22773061

Continuously Moving Objects Framework to Monitor Result Changes of Spatio Temporal Queries using Data Stream Management System

Brijesh Panday, Vinod Kumar Yadav, Shreya Paul
Computer Science and Engg. Department, GITM, Lucknow, India
brijesh84academics@gmail.com
Information Technology Department, SRMCEM, Lucknow, India
vkynita@gmail.com
Information Technology Department, SRMCEM, Lucknow, India
sp.shreyapaul01@gmail.com

Abstract— In this paper the concept of safe limit in the framework for continuously moving objects to monitor result changes of spatio temporal queries has been proposed. In this framework the movement of the moving object is monitored with the help of user defined aggregate function over spatio temporal reference in data stream management system. The expected movement of the moving object is plotted on the graph and the spatio temporal queries are answered on the basis of that, until and unless the difference between the expected movement and actual movement is more than the safe limit. This makes the framework more efficient than the previously given framework. The safe limit can be any range with respect to space and time of a moving object which varies according to different parameters such as size of the object, velocity with which it is moving and etc.

Keywords: - Data stream, Safe limit, Actual movement, Expectant movement, Spatio temporal query.

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 7, No 3
editor@cirworld.com
www.cirworld.com, member.cirworld.com

I. INTRODUCTION

The general idea behind moving objects databases is to represent moving entities in databases and ask queries about them. Moving entities could be people, animals, vehicles, spacecrafts etc. For these moving entities only the time-dependent position in space is relevant, not the extent, therefore we can characterize them as moving points [2]. There are also moving entities with an extent, such as, hurricanes, forest fires, epidemic dis-eases, oil spills, armies, and etc. These can be characterized as moving regions [2]. Extending database technology to deal with such objects would require to provide facilities in a DBMS data model that not only describe such entities but to extend the query language by constructs for analyzing them as well. Spatio temporal queries deal with spatial characteristics or relationships along time [7]. It returns spatial components or values at a specific time or valid time of moving objects. A spatio temporal query involves the intersection operation like which objects intersect with object o at a given point or within a given area at a given time instant or during a given time period [1]. Main objective of this paper is to reduce the cost of location updates so as to increase the efficiency of the framework for monitoring moving objects. And to monitor result changes of spatial temporal queries as objects move and monitor them as accurately as possible.

A Unified approach based on spatio temporal data type constructor can be used to develop such a framework. Using spatial and time instant data types separately for representing spatio temporal characteristics of data streams is not that good for such a framework. This type constructor along with a spatial type representing object location produces a corresponding type whose values are pairs consisting of an instant of time and a value of spatial type. This approach is feasible to represent temporal development of an object location within one time interval by using previous values of this temporal data type.

The monitoring system framework consists of three basic building blocks i.e. Efficiency, Accuracy and Privacy. And to satisfy the measures of these building blocks is of much importance. For efficiency the framework significantly reduces location updates to only when an object is crossing the safe limit as prescribed with respect to the object. For accuracy the framework offers correct monitoring result at any time as opposed to only at the time when the object just cross the safe limit. For privacy the concept of safe limit again comes into the frame. The client may not want to expose their exact point location to the database server to avoid spatio temporal correlation inference attack by which an adversary may infer user's private information. Therefore, the object may not be exactly located if safe limit concept is used. The safe limit can be defined as a limit within which the change of object location does not disqualify the graph made by the relations derived of the moving object. And hence does not change the result of any registered spatio temporal query.

II. RELATED WORK

Moving object databases in which objects or queries or both of them move continuously has got a much attention in the recent past. Many frameworks have been designed and are being studied in contention with this. There are two different perspectives in the field of moving object database. The first one considers the current and future movement while the second one is based on the modeling of the history of movement.

The research work done by guiting and colleagues presented in [1, 2] is a great contribution in spatial temporal database community. They developed an in-depth and formal theory for moving objects, providing formal foundation to any further work on spatial temporal databases and/or moving object. They have taken two types of data for objects. On the basis of these data they evaluate the queries.

Wolfson and colleagues [17, 18] focused on trajectory location management. It captures the current movement of spatial objects and their anticipated locations in near future. They have used a data model called MOST that supports in determining the current and expected future movement using dynamic attributes. Dynamic attributes implicitly change their value over time without explicit updates. Moving point objects are restricted by this approach.

Much work has been done in the area of data stream within several projects each producing a prototype DSMS. In [5, 19] Stanford transformed the streams into relations using window operators and then queried using standard relational operators. Again then transforms the results back into streams. In [20, 21] Mill focused on data mining streaming data, that enables the user to define custom aggregates which are then used to process streaming data. UC Berkley project on Telegraph enables it to make scheduling decisions for each tuple. This framework is implemented as an extension of PostgeSQL DBMS in C/C++ programming language.

Haibo hu, jianliang xu and dik lun lee have added a great value to the research of moving object by developing the privacy aware monitoring framework in [6] and [9]. Emir meskovic, Zdravko galic and Mirta baranovic contribution in [7] and [8] is significant in managing moving objects with context to data stream management system.

In paper [16] author describes the two different SQL extensions for data streams and its associated semantics. Tuple-based execution provides a way to react to primitive events as soon as they are seen by the system while Timebased execution provides a way to model simultaneity. Its result gives the user the control over the granularity at which one can express simultaneity. Uncertainty and privacy issues are the hot topic of studies in moving object monitoring. To protect location privacy, various cloaking or anonymizing techniques have been proposed to hide the client actual location. Among them are spatiotemporal cloaking [16], the clique-cloak [10, 11], the Casper anonymizer [15], hilbASR [13, 14] and peer-to-peer cloaking [12].

Rest of the paper organized as follows: Section II defines the problem statement. Section III describes the functioning of framework. Section IV presents the location update strategy. Section V gives some example of spatio temporal queries. Section VI shows the experimental results. Finally Section VII concludes and outlines future work.

ISSN 22773061

III. PROBLEM STATEMENT

In moving object database management system where objects move continuously and queries related to them also changes simultaneously. The purpose is to design an efficient framework to monitor the result changes of spatial temporal queries as objects move using data stream management system. Accuracy and privacy are also the issues that need to be considered in framework.

IV. FUNCTIONING OF THE FRAMEWORK

The framework basically consist of two parts the database server and the application server. It is assumed that the numbers of moving objects are much more than number of spatial temporal queries. Hence all the registered queries can be made fit into the main memory of the database server whereas number of moving objects cannot. Location updates are handled sequentially at the database server to avoid read/write consistency complicacy. Communication cost between every client and server is assumed to be constant. Application server can register spatial temporal queries to the database server at any time.

The location manager at the database server computes the motion of the moving object using user defined aggregate function and accordingly updates the position of the moving object at the object index. The query processor identifies the queries those are affected by the new position of the moving object using the query index and evaluates them even before the object reaches that position in actual. The location manager on the other hand also computes the safe limit of the moving object at a certain time interval and informs it to the location updater. When the moving object crosses the safe limit the location updater updates it to the location manager which in turn updates the object index and the queries are reevaluated accordingly. It is highly impossible for the moving object to cross the safe limit as user defined aggregate function computes nearly exactly the motion of the moving object. We have used data stream management system to evaluate the spatial temporal queries. Data stream represents potentially unbounded sequence of elements < s, t > where s is tuple with schema s and t is timestamp of an element. For any element < s1, t1 > that precedes element < s2, t2 > in data stream it holds that t1 ≥ t2. Each data stream element is mapped to exactly one timestamp that represents its unique time reference. For every moving object spatial stamp 'ϭ' is also attached to each data stream element. It represents a location and geometry of a data source when element is generated. Time stamp and spatial stamp together uniquely determine spatio temporal status of a moving object and are explicitly included in the schema of data stream [12].

A. User defined aggregate function (UDAF) in DSMS:

For the creation of the user defined aggregate function the user has to implement three functions namely Init, Accumulate and Terminate. These functions are defined as follows:

1. Init- It is used to initialize the variables needed for the computation later on.

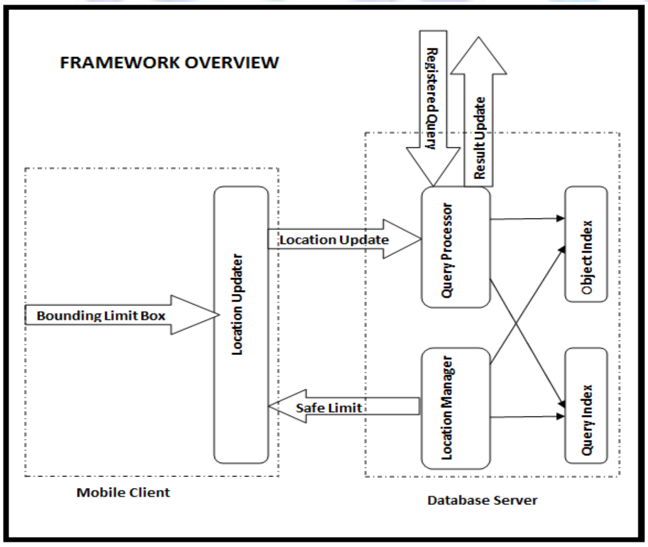


Figure 1: Shows the overview of framework.

2. Accumulate- It is called once for each value aggregated. This function adds the value to the running total computed so far.
3. Terminate- This function is used to end the calculation and return the final value of the aggregate function.

UDAF in which terminate operation is not defined are called non blocking and can be applied to data stream freely.

B. UDAF over spatio temporal references:

The sliced representation of the moving object is obtained by the user defined aggregate function over spatio temporal reference of data stream elements in data stream management system. With the help of different window operators i.e. stream to relation operators on spatio temporal data stream proposed UDAF produce moving objects in a form of sliced representation as

ISSN 22773061

intermediate result of continuous query. Efficient management of moving objects compressed in a spatio temporal data stream is enabled by implementing algorithms for operation over moving object database context. DSMS that support not only data stream but persistent relation as well can be extended with ability to store intermediate result of continuous queries, i.e. moving objects gained by using UDAF, into persistent relation[8]. With this approach history of movements of moving objects extracted out of spatio temporal data stream can be stored in persistent relation. A powerful set of previously unavailable continuous queries can be evaluated by joining relations containing history of movement with data streams carrying information about current movements.

V. LOCATION UPDATES STRATEGY

The location updater updates the location of objects only when its deviation is more than prescribed safe limit. The safe limit can be any range which can be defined as per the size, speed or various other parameters of moving object.

A. Notations and Descriptions:

DSMS - Data stream management system.

UDAF- User defined aggregate function.

EM - Expected movement of the moving object.

AM - The actual movement of the moving object.

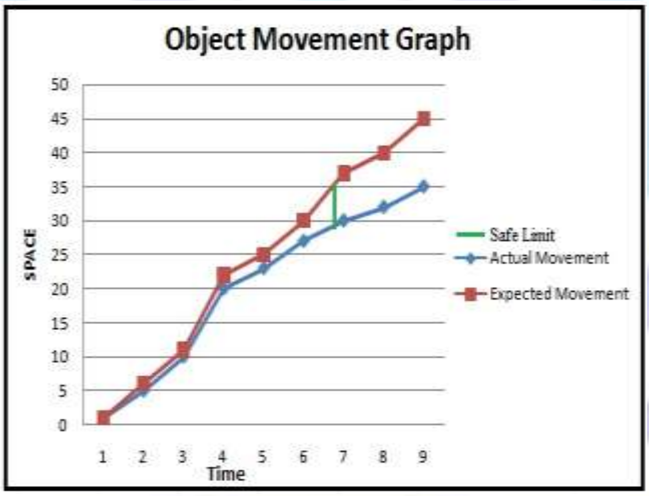
SL - Safe limit for moving object.

1. AM ~ EM < SL

In this case the spatio temporal queries are answered only with the help of expected movement of the moving object irrespective of the actual movement.

2. AM ~ EM ≥SL

In this case the spatio temporal queries are answered only by the actual movement and for future new safe limit are calculated and the graph is plotted according to that movement.



Graph 1:- Shows the safe limit for object movement.

In the given above graph expected movement is plotted on the basis of relation created by the history of movement of the moving object. The actual movement of the object is notified and until and unless the object crosses the safe limit the spatio temporal queries are answered on the basis of expected movement. Safe limit of the object depends upon various parameters such as size of the object, velocity with which it is moving, number of moving objects in that region and etc.

VI. SOME EXAMPLES OF SPATIO TEMPORAL QUERIES THAT CAN BE SOLVED USING THIS APPROACH

Q1: Find all buses found within the area of interest (e.g. arectangle) sometimes during last 20 minutes. SELECT id, carType FROM busStream [RANGE 20 minutes'] WHERE busType = 'BUS' GROUP BY id, busType HAVING passes(moving(pos), STGeom FromText ('POLYGON (575000.0 322000.0, 557000.0 323000.0)', 25832).

Q2: For each police jeep, find its minimal distance from the point of interest during last half an hour. SELECT id, val (initial (atmin (distance (moving (pos), STGeom FromText ('POINT(557234 322701)', 25832)))) FROM jeepStream [RANGE '30 minutes'] WHERE vType = 'POLICE' GROUP BY id.

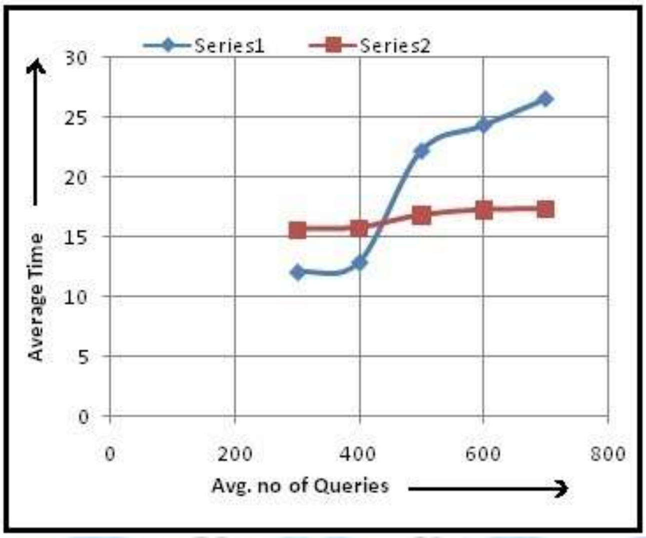
Q3: Find each time instant when policeman A was heading north (let's say, within 10 degrees of the exact north direction) during last half an hour. SELECT deftime(at (mdirection (moving (pos), range (80, 100))) FROM carStream [RANGE '24 hours'] WHERE carType = 'POLICE' AND driver = 'A' GROUP BY id.

Q4: Find all drives that have crossed more than 50 km during last two hours. SELECT id, driver FROM carStream [RANGE '2 hour'] WHERE carType = 'BUS' GROUP BY id, driver HAVING length (trajectory (moving (pos))) > 50/1000.

For the implementation of the proposed approach we have used the java code. Some data has been collected with respect to space and time of a moving object. On the basis of the history of movement the safe limit for the moving object is calculated. With the help of safe limit we predicted the movement of the moving object. Hence in this way we find the expected movement of the moving object. The spatial temporal queries are answered on the basis of this expected movement until and unless the moving object crosses its safe limit. Using this concept the efficiency of the framework is increased which we have shown in the given graph. Increase in efficiency makes it better than the previous given framework.

A. Efficiency:-

Efficiency of the system will increase (Graph-2). Because in previous models when an object sends a location update, the query processor determines those queries that are affected by this update using query index. Then it evaluates them by using the object index and updated query results are reported to servers who register these queries. On the basis of indexes, it sends back as a response to the object. But in new approach all these steps are not followed completely at all. Updated values are compared with safe limit, if it is greater than safe limit then this new value will send back as a response to the object. Otherwise respond back on the basis of previous values and increase efficiency of the system.



Graph 2:- Shows the result of object movement.

B. Accuracy:-

Accuracy of the system will decrease because we response for some queries (in case when updated value is less than safe limit) on the basis of previous values of the objects that are approximate values.

C. Privacy:-

Here privacy is inversely proportional to accuracy. Because most of the time we respond for queries on the basis of previous values, and these values are approximate. So it will tough for attacker to get exact position of objects that increase the privacy.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we proposed the concept of safe limit in the framework for continuously moving objects to monitor result changes of spatio temporal queries. In this framework the movement of the moving object is monitored with the help of of user defined aggregate function over spatio temporal reference in data stream management system. The expected movement of the moving object is plotted on the graph and the spatio temporal queries are answered on the basis of that, until and unless the difference between the expected movement and actual movement is more than the safe limit. This makes the framework more efficient than the previously given framework. For the future work we can enhance the concept of safe limit by calculating some formula on the basis of which we can calculate the safe limit for every moving object. Different parameters and the ratio in which they contribute for the calculation of the safe limit can be studied and implemented.

REFERENCES

[1] M. Erwig, R. H. Gting, M. Schneider, M. Vazirgiannis, Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases, Revised Version, 1999.

[2] L. Forlizzi, R. H. Gting, E. Nardelli, M. Schneider, A Data Model and Data Structures for Moving Object Databases, Proc. of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp.319-330.

[3] R. H. Gting, M. Schneider, Moving Objects Databases Morgan Kaufmann 2005.

[4] S. Cohen, User-Defined Aggregate Functions: Bridging Theory and Practice, Proceedings of the 2006 ACM SIGMOD International Conference on Management Data, 2006, pp. 49-60.

[5] A. Arasu, S. Babu, J. Widom: The CQL continuous query language: semantic foundations and query execution, The International Journal on Very Large Databases, 15(2), 121-142.

[6] Haibo Hu, Jianliang Xu and Dik Lun Lee, PAM: A n Efficient and Privacy-A ware Monitoring Framework for Continuously Moving Objects, IEEE Transaction on knowledge and data engineering, vol 22, no.3, March 2010.

[7] Kresmeric Krizanovic, Zdravko Galic and Mirta Baranovic, Spatio Temporal Data Streams: An Approach to Managing Moving Objects, MIPRO 2010. May 24-28, 2010, opatijia, Croatia.

[8] Kresmeric Krizanovic, Zdravko Galic and Mirta Baranovic, Managing Moving objects in the context of spatio temporal data streams, MIPRO 2011. May 23-27, 2011, opatijia, Croatia.

[9] H. Hu, J. Xu and D. L. Lee, A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects, Proc. ACM SIGMOD, pp. 479-490, 2005.

[10] B. Gedik and L. Liu, Location Privacy in Mobile Systems: A Personalized Anonymization Model, Proc. IEEE Intl Conf. Distributed Computing Systems (ICDCS), pp. 620-629, 2005.

[11] B. Gedik and L. Liu, Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms, IEEE Trans. Mobile Computing, vol. 7, no. 1, pp. 1-18, Jan. 2008.

[12] S. Cohen, User-Defined Aggregate Functions: Bridging Theory and Practice, Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, 2006, pp. 49-60.

[13] G. Ghinita, P. Kalnis and S. Skiadopoulos, Prive: Anonymous Location Based Queries in Distributed Mobile Systems, Proc. Intl World Wide Web Conf. (WWW 07), pp. 371-380, 2007.

[14] P. Kalnis, G. Ghinita, K. Mouratidis and D. Papadias, Preventing Location-Based Identity Inference in Anonymous Spatial Queries, IEEE Trans. Knowledge and Data Eng., vol. 19, no. 12, pp. 1719-1733, Dec. 2007.

[15] M. F. Mokbel, C. -Y. Chow and W. G. Aref, The New Casper: Query Processing for Location Services without Compromising Privacy, Proc. Intl Conf. Very Large Data Bases (VLDB), pp. 763-774, 2006.

[16] M. Gruteser and D. Grunwald, Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking, Proc.MobiSys, 2003.

[17] A. P. Sistla, O. Wolfson, S. Chamberlain and S. Dao. Modeling and querying moving objects. In Proceedings of the 13th International Conference on Data Engineering, pages 422-432, 1997.

[18] O. Wolfson, B. Xu, S. Chamberlain and L. Jiang. Moving objects databases: Isssues and solutions. In Proceedings of the 10th International Conference on Scientific and Statistical Database Management, pages 111-122, 1998.

[19] The STREAM Group. STREAM: The Stanford Stream Data Manager. IEEE Data Engineering Bulletin, 2003.

[20] Y. Bai, H. Thakkar, C. Luo, H. Wang, C. Zaniolo, ”A Data Stream Language and system Designed for Power and Extensibility”, Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pages 337-346, 2006.

[21] Y. N. Law, H. Wang, C. Zaniolo, ”Query Languages and Data Models for Database Sequences and Data Streams”,

Proceedings of the 30th International Conference on Very Large Databases Volume 30, pages 492-503, 2004.